IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Mathew C. Mattina et al.	Examiner: Charles Anya
Serial No.:	09/924,934	Group Art Unit: 2194
Filed:	August 8, 2001	Docket No.: 200302133-1
Title:	Mechanism for Handling Load Lock/Store Conditional Primitives in Directory-Based Distributed Shared Memory Multiprocessors	

APPEAL BRIEF UNDER 37 C.F.R. § 41.37

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This Appeal Brief is filed in response to the Final Office Action mailed July 26, 2006 and Notice of Appeal filed on October 25, 2006.

AUTHORIZATION TO DEBIT ACCOUNT

It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's deposit account no. 08-2025.

I. REAL PARTY IN INTEREST

The real party-in-interest is the assignee, Hewlett-Packard Development Company, a Texas Limited Partnership, having its principal place of business in Houston, Texas. Assignments are recorded at Reel/Frame 012067/0828 and 014628/0103.

II. RELATED APPEALS AND INTERFERENCES

There are no known related appeals or interferences known to appellant, the appellant's legal representative, or assignee that will directly affect or be directly affected by or have a bearing on the Appeal Board's decision in the pending appeal.

III. STATUS OF CLAIMS

Claims 1 – 34 stand finally rejected. The rejection of claims 1 – 34 is appealed.

IV. STATUS OF AMENDMENTS

No amendments were made after receipt of the Final Office Action. All amendments have been entered.

V. SUMMARY OF CLAIMED SUBJECT MATTER

The following provides a concise explanation of the subject matter defined in each of the claims involved in the appeal, referring to the specification by page and line number and to the drawings by reference characters, as required by 37 C.F.R. § 41.37(c)(1)(v). Each element of the claims is identified by a corresponding reference to the specification and drawings where applicable. Note that the citation to passages in the specification and drawings for each claim element does not imply that the limitations from the specification and drawings should be read into the corresponding claim element or that these are the sole sources in the specification supporting the claim features.

Paragraph [0017] in the Brief Summary of the Invention section describes a multiprocessor distributed shared memory system in which a processor may obtain exclusive control of a data block by asserting a Load Lock signal to the Home processor that controls that data block If the processor with exclusive control does not complete

operations on the data block prior to the time that the data block is displaced from the cache of that processor, it issues a Victim To Shared message, thereby indicating to the Home processor that it should remain a sharer of the data block, and the data block is written back. In the event that another processor seeks exclusive rights to the same data block, the Home processor issues an Invalidate message to what was previously the Owner processor. This causes that processor to recognize that the Load Lock/Store Conditional operation pair has failed. If no Invalidate message is received, and when the processor executes the Store Conditional instruction, it seeks exclusive control of the data block by issuing a Read with Modify Intent Store Conditional message to the Home processor. If that processor is still a sharer, a writeable copy of the data block is sent to that processor, who completes modification of the data block. If the Home processor does not recognize that processor as a sharer of that data block, this indicates to the Home processor that an intervening write has occurred, and the Home processor returns a Store Conditional Failure message.

Claim 1 recites a distributed multiprocessing computer system, which includes a plurality of processors each coupled to an associated memory module, wherein each of the associated memory modules may store data that is shared between said processors (Fig. 2, ¶30, lines 1-9). The system comprises a Home processor that includes a memory block and a directory for said memory block in the associated memory module (¶26, lines 11-13); an Owner processor that includes a cache memory, and wherein said Owner processor obtains an exclusive copy of said memory block, and stores said exclusive copy of said memory block in said cache memory (¶26, lines 13-14); and wherein said Owner processor may displace the exclusive copy of said memory block, and return said displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor remains a sharer of said memory block (¶17, lines 1-9; ¶36, lines 1-12).

Claim 2 recites the distributed multiprocessing computer system of claim 1, wherein said Owner processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the Home processor indicates that said Owner processor has obtained exclusive control of said memory block (¶s 17, 18, and 37).

Claim 4 recites a distributed multiprocessing computer system of claim 3, wherein said Owner processor includes a register in which an address is stored representing the memory block obtained in response to the Load Lock instruction (¶19), and wherein said Owner processor compares the address of any displaced data with the address stored in said register (Fig. 3; ¶40).

Claim 5 recites the distributed multiprocessing computer system of claim 4, wherein the Owner processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register (Fig. 3; ¶36 and ¶40, lines 7-12).

Claim 13 recites a method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors (Fig. 2; ¶30, lines 1-9). The method comprises the acts of: requesting a copy of a memory block from a Home processor to perform a write operation on the copy of the memory block (¶18, lines 4-6); storing said copy of said memory block exclusively in a cache memory associated with an Owner processor (¶18, lines 4-7); updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor (¶32); displacing said copy of said memory block from said cache memory prior to completion of operations on said memory block (¶17, lines 3-7); transmitting a message to said Home processor relinquishing exclusive control of said memory block, while indicating that said Owner processor should still be deemed a sharer of said memory block (¶17, lines 3-7; ¶36, lines 3-12).

Claim 14 recites the method of claim 13, wherein the copy of the memory block is requested using a Load Lock instruction from the Owner processor to the Home processor (¶s 17, 18, and 37).

Claim 15 recites the method of claim 13, wherein the Load Lock instruction forms part of a Load Lock/Store Conditional instruction pair (¶s 17 and 38).

Claim 17 recites the method of claim 13, wherein the act of displacing said copy of said memory block includes comparing the address of any displaced memory block with an address of any memory block for which an exclusive copy resides in the Owner processor (Fig. 3; ¶40).

Claim 18 recites the method of claim 17, wherein the act of transmitting a message includes assertion of a Victim To Shared message if the address of the displaced memory block matches the address of any memory block for which an exclusive copy resides in the Owner processor (Fig. 3; ¶40, lines 7-12), and wherein the directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message (Fig. 3; ¶36 and ¶40, lines 7-12).

Claim 24 recites a distributed multiprocessing computer system (Fig. 2; ¶30, lines 1-9). The system comprises a first processor that includes a memory block and a directory associated with said memory block that tracks the status of said memory block (Fig. 2, ¶32, lines 1-3); a second processor that includes a cache memory, and wherein said second processor is capable of requesting an exclusive copy of said memory block that is stored in said cache memory (¶26, lines 13-14); and wherein said second processor may displace the exclusive copy of said memory block prior to completing processing of said memory block (¶17, lines 1-9), and said second processor transmits a signal to said first processor indicating that said second processor relinquishes exclusive control of said memory block but should remain a sharer of said memory block (¶36, lines 1-12).

Claim 25 recites the distributed multiprocessing computer system of claim 24, wherein said second processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the first processor indicates that said second processor has obtained exclusive control of said memory block (¶s 17, 18, and 37).

Claim 27 recites the distributed multiprocessing computer system of claim 24, wherein said second processor includes a register in which an address is stored representing the memory block exclusively obtained from said first processor, and wherein said second processor compares the address of any displaced data with the address stored in said register. (Fig. 3; ¶40)

Claim 28 recites the distributed multiprocessing computer system of claim 27, wherein the second processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register (Fig. 3; ¶36 and ¶40, lines 7-12).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1, 2, 13, 14, 16, 24, and 25 are rejected under 35 USC § 102(b) a being anticipated by USPN 5,887,138 (Hagersten).

Claims 3 and 26 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing).

Claim 15 is rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,425,050 (Beardsley).

Claims 4, 17, and 27are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing) and USPN 5,937,199 (Temple).

Claims 5-7, 18, 19, 28, and 29 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing) and USPN 5,937,199 (Temple) and US application number 2001/0010068 (Michael).

Claims 8-12, 20-23, and 30-34 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing) and USPN 5,937,199 (Temple) and US application number 2001/0010068 (Michael) and USPN 6,425,050 (Beardsley).

VII. ARGUMENT

The rejection of claims 1 – 34 is improper, and Applicants respectfully requests withdraw of this rejection.

The claims do not stand or fall together. Instead, Applicants present separate arguments for various independent and dependent claims. Each of these arguments is separately argued below and presented with separate headings and sub-heading as required by 37 C.F.R. § 41.37(c)(1)(vii).

Claim Rejections: 35 USC § 102(b)

Claims 1, 2, 13, 14, 16, 24, and 25 are rejected under 35 USC § 102(b) a being anticipated by USPN 5,887,138 (Hagersten).

In order to anticipate a claim, the reference must teach every element in the claim (see MPEP 2131). Applicants respectfully traverse the rejections because each of the independent claims recites at least one element that is not disclosed in Hagerston.

By way of example, claim 1 recites the following:

> wherein said Owner processor may displace the exclusive copy of said memory block, and return said displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor remains a sharer of said memory block.

As recited in claim 1, the Owner processor receives an exclusive copy of data from a memory block of the Home processor. The Owner processor then displaces this exclusive copy and returns it to the Home processor. When the exclusive copy of the data is returned to the Home processor, a signal instructs the Home processor that the Owner processor **remains a sharer of the memory block**. In other words, the Owner processor is able to displace data from memory of the Home processor while maintaining a role that it does not in fact have (i.e., the role of a processor that continues to hold a copy of data in memory). Hagerston does not disclose these claim recitations.

The Examiner cites Hagerston at column 13, lines 21 – 39. In contrast to claim 1, this section of Hagerston defines coherency states. Applicants believe that the Examiner is mixing different definitions of different coherency states in an attempt to show the claim recitations. This section of Hagerston states (emphasis added):

> In one embodiment, the coherency states employed by computer system 10 are **modified, owned, shared,** and **invalid**. The **modified state** indicates that the SMP node 12 has updated the corresponding coherency unit. Therefore, other SMP nodes 12 do not have a copy of the coherency unit. Additionally, when the modified coherency unit is discarded by the SMP node 12, the coherency unit is stored back to the home node. The **owned state** indicates that the SMP node 12 is responsible for the coherency unit, but other SMP nodes 12 may have shared copies. Again, when the coherency unit is discarded by the SMP node 12, the coherency unit is stored back to the home node. The **shared state** indicates that the SMP node 12 may read the coherency unit but may not update the coherency unit without acquiring the owned state. Additionally, other SMP nodes 12 may have copies of the coherency unit as well. Finally, the **invalid state** indicates that the SMP node 12 does not have a copy of the coherency unit. In one embodiment, the modified state indicates write permission and any state but invalid indicates read permission to the corresponding coherency unit.

Nowhere does this section of Hagerston teach or even suggest that when a first processor displaces exclusive copy of data from memory of a second processor, the first processor returns the data and a signal indicating that the second processor remains a sharer of a memory block storing the data.

Independent claim 13 recites "displacing said copy of said memory block from said cache memory … [and] transmitting a message to said Home processor relinquishing

exclusive control of said memory block, while indicating that said Owner processor should still be deemed a sharer of said memory block." Hagerston does not disclose these recitations.

The Examiner relies on the Hagerston at column 13, lines 21 – 39 for allegedly disclosing these recitations. As noted above in connection with claim 1, the cited section of Hagerston merely defines four different coherency states (i.e., modified, owned, shared, and invalid). This section of Hagerston, however, does not disclose that a processor retains a role as sharer after having displaced the data from memory. Specifically, claim 13 recites that the Owner processor displaces the memory block from its cache and transmits a message to the Home processor. The message **(1)** relinquishes exclusive control of the memory block and **(2)** indicates that the Owner processor is still deemed a shared of the memory block. Nowhere does the cited section of Hagerston state that a first processor transmits a message to a second processor to relinquish exclusive control of second processor's memory block and to indicate the first processor should still be deemed a shared of the second processor's memory block.

Independent claim 24 recites a second processor that receives an exclusive copy of a memory block of a first processor. The second processor displaces the exclusive copy of the memory block of the first processor and then "transmits a signal to said first processor indicating that said second processor relinquished exclusive control of said memory block **but should remain a sharer of said memory block**" (emphasis added). Hagerston does not disclose these recitations.

The Examiner relies on the Hagerston at column 13, lines 21 – 39 for allegedly disclosing these recitations. As noted above, the cited section of Hagerston merely defines four different coherency states (i.e., modified, owned, shared, and invalid). This section of Hagerston, however, does not disclose that a processor retains a role as sharer after having displaced the data from memory. Specifically, claim 24 recites that the second processor receives an exclusive copy of data from a memory block of the first processor. The second processor displaces the memory block from its cache and transmits a signal to the first processor. The signal indicates "that said second processor relinquishes exclusive control of said memory block but should remain a sharer of said memory block." Nowhere does the cited section of Hagerston disclose these recitations.

Claims 2, 14, and 25 are grouped together; and claim 2 is selected for discussion. Claim 2 recites that the Owner processor obtains exclusive control with a specific type of instruction called a "Load Lock instruction." Nowhere does Hagerston teach these recitations.

The Examiner cites Hagerston at column 19, lines 15-34. This section of Hagerston discloses that a requesting agent transmits a "read to own" request to a home node. Hagerston explains that a "read to own operation causes transfer of the requested coherency unit to the requesting node. The requesting node receives the coherency unit in the modified state" (see column 19, lines 10-13). Nowhere does Hagerston state that the requesting agent obtains exclusive control using a Load Lock instruction.

Applicants respectfully argue that anticipation under section 102 can be found only if a single reference shows exactly what is claimed (see *Titanium Metals Corp. v. Banner*, 778 F.2d 775, 227 U.S.P.Q. 773 (Fed. Cir. 1985)).

Claim Rejections: 35 USC § 103(a)

Claims 3 and 26 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing). These rejections are traversed.

Claim 3 depends from claims 2 and 1; and claim 26 depends from claim 24. Doing fails to cure the deficiencies of Hagersten argued above in connection with claims 1, 2, and 24. Thus, for at least the reasons provided above, claims 3 and 26 are allowable over Hagersten and Doing.

Claim Rejections: 35 USC § 103(a)

Claim 15 is rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,425,050 (Beardsley).

Claim 15 recites a Load Lock instruction that "forms part of a Load Lock/Store Conditional instruction pair." The Examiner admits that Hagerston does not teach this recitation (see Final OA at p. 5). The Examiner, however, attempts to cure this deficiency with Beardsley at column 4, lines 38-67.

Beardsley at column 4, lines 38-67 teaches writes and destages can require exclusive access to tracks in a cache. This section also discusses that read requests are serviced while user data and metadata are destaged. Nowhere does this section of Beardsley teach that a specific instruction called a Load Lock instruction "forms part of a Load Lock/Store Conditional instruction pair." Beardsley does not suggest such instructional pairs.

Claim Rejections: 35 USC § 103(a)

Claims 4, 17, and 27are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing) and USPN 5,937,199 (Temple).

Claims 4, 17, and 27 are grouped together; and claim 4 is selected for discussion. Claim 4 recites the following recitation:

> said Owner processor includes a register in which an address is stored representing the memory block obtained in response to the Load Lock instruction, and wherein said Owner processor compares the address of any displaced data with the address stored in said register.

The Examiner admits that Hagersten and Doing do not teach these recitations (see Final OA at p. 6). Applicants agree with this admission. The Examiner, however, attempts to cure these deficiencies with Temple at column 10, lines 5-36). Applicants respectfully disagree.

Temple at column 10, lines 5 – 36 discusses a specific type of atomic execution known as a "read-modify-write" operation. Temple discusses how these read-modify-write operations are executed "to ensure that the storage location from which the data is accessed is not subsequently accessed by the system." Nowhere does Temple disclose or even suggest that read-modify-write operations utilize a processor that stores an address representing a memory block obtained from another processor in response to a "Load Lock instruction." Further, claim 4 requires that the processor compare "the address of

any displaced data with the address stored in said register." **Temple never discusses or even suggests such a comparison.**

Claim Rejections: 35 USC § 103(a)

Claims 5-7, 18, 19, 28, and 29 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing) and USPN 5,937,199 (Temple) and US application number 2001/0010068 (Michael).

Claims 5, 18, and 28 are grouped together; and claim 5 is selected for discussion. Claim 5 recites that the Owner processor asserts a specific type of message, called a "Victim To Shared message." This message is asserted "if the address of any displaced data matches the address stored in said register." The Examiner admits that Hagersten, Doing, and Temple do not teach this recitation. Applicants agree with these admissions. The Examiner, however, attempts to cure these deficiencies with Michael at paragraph [0014]. Applicants strongly disagree.

Paragraph [0014] in Michael is reproduced below:

> [0014] It is an object of the invention to provide a cache system that minimizes directory cache pollution, leading to higher directory cache hit ratios and resulting improvement in system performance.

Where does Michaels discuss or suggest a specific type of message, called a "Victim To Shared message" in this section? Michaels does not. Where does Michaels discuss or suggest that a Victim To Shared Message is asserted "if the address of any displaced data matches the address stored in said register" as recited in claim 5? Michaels does not.

Claim Rejections: 35 USC § 103(a)

Claims 8-12, 20-23, and 30-34 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,887,138 (Hagersten) in further view of USPN 6,438,671 (Doing) and USPN 5,937,199 (Temple) and US application number 2001/0010068 (Michael) and USPN 6,425,050 (Beardsley).

Claims 8-12, 20-23, and 30-34 are allowable for at least the reasons provided above with respective independent claims 1, 13, and 24 and various intervening dependent claims.

CONCLUSION

In view of the above, Applicants respectfully request the Board of Appeals to reverse the Examiner's rejection of all pending claims.

Any inquiry regarding this Amendment and Response should be directed to Philip S. Lyren at Telephone No. 832-236-5529. In addition, all correspondence should continue to be directed to the following address:

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, Colorado 80527-2400

Respectfully submitted,

/Philip S. Lyren #40,709/

Philip S. Lyren
Reg. No. 40,709
Ph: 832-236-5529

VIII. Claims Appendix

1. (Previously presented) A distributed multiprocessing computer system, which includes a plurality of processors each coupled to an associated memory module, wherein each of the associated memory modules may store data that is shared between said processors, said system comprising:

a Home processor that includes a memory block and a directory for said memory block in the associated memory module;

an Owner processor that includes a cache memory, and wherein said Owner processor obtains an exclusive copy of said memory block, and stores said exclusive copy of said memory block in said cache memory; and

wherein said Owner processor may displace the exclusive copy of said memory block, and return said displaced copy of said memory block to said Home processor with a signal indicating that said Owner processor remains a sharer of said memory block.

2. (Original) The distributed multiprocessing computer system of claim 1, wherein said Owner processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the Home processor indicates that said Owner processor has obtained exclusive control of said memory block.

3. (Original) The distributed multiprocessing computer system of claim 2, wherein said Owner processor is capable of executing multiple threads concurrently, and may displace data associated with a non-executing thread from its associated cache memory.

4. (Original) The distributed multiprocessing computer system of claim 3, wherein said Owner processor includes a register in which an address is stored representing the memory block obtained in response to the Load Lock instruction, and wherein said Owner processor compares the address of any displaced data with the address stored in said register.

5. (Original) The distributed multiprocessing computer system of claim 4, wherein the Owner processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register.

6. (Original) The distributed multiprocessing computer system of claim 5, wherein the Owner processor asserts a Victim message if the address of any displaced data does not match the address stored in said register.

7. (Original) The distributed multiprocessing computer system of claim 5, wherein the directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

8. (Original) The distributed multiprocessing computer system of claim 7, wherein said Owner processor subsequently re-obtains an exclusive copy of said memory block to complete execution of the non-executing thread.

9. (Original) The distributed multiprocessing computer system of claim 8, wherein the Owner processor asserts a Read-with-Modify Intent Store Conditional instruction to the Home directory to again request an exclusive copy of said memory block.

10. (Original) The distributed multiprocessing computer system of claim 9, wherein, in response to the Read-with-Modify Intent Store Conditional instruction, the Home directory determines if the Owner processor is a sharer of the memory block, and if so, the Home directory sends an exclusive copy of the memory block to the Owner processor.

11. (Original) The distributed multiprocessing computer system of claim 10, wherein the Home directory invalidates all other sharers when it sends an exclusive copy of the memory block to the Owner processor.

12. (Original) The distributed multiprocessing computer system of claim 9, wherein the Home directory determines if the Owner processor is a sharer of the memory block, and if not, the Home directory sends a Store Conditional Failure message to the Owner processor.

13. (Original) A method of maintaining memory coherence in a distributed shared memory computer system including a plurality of processors, comprising the acts of:

requesting a copy of a memory block from a Home processor to perform a write operation on the copy of the memory block;

storing said copy of said memory block exclusively in a cache memory associated with an Owner processor;

updating a coherence directory for said memory block in said Home processor to indicate the write operation on the copy of said memory block in said cache memory associated with the Owner processor;

displacing said copy of said memory block from said cache memory prior to completion of operations on said memory block;

transmitting a message to said Home processor relinquishing exclusive control of said memory block, while indicating that said Owner processor should still be deemed a sharer of said memory block.

14. (Original) The method of claim 13, wherein the copy of the memory block is requested using a Load Lock instruction from the Owner processor to the Home processor.

15. (Original) The method of claim 13, wherein the Load Lock instruction forms part of a Load Lock/Store Conditional instruction pair.

16. (Original) The method of claim 13, wherein the act of updating the coherence directory includes modifying a register to indicate that the Owner processor has an exclusive copy of the memory block.

17. (Original) The method of claim 13, wherein the act of displacing said copy of said memory block includes comparing the address of any displaced memory block with an address of any memory block for which an exclusive copy resides in the Owner processor.

18. (Original) The method of claim 17, wherein the act of transmitting a message includes assertion of a Victim To Shared message if the address of the displaced memory block matches the address of any memory block for which an exclusive copy resides in the Owner processor, and wherein the directory associated with the Home processor indicates that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

19. (Original) The method of claim 18, further comprising the act of updating the coherence directory to indicate that said Owner processor has become a sharer of said memory block in response to said Victim To Shared message.

20. (Original) The method of claim 13, further comprising the act of asserting a request to again obtain an exclusive copy of said memory block.

21. (Original) The method of claim 20, wherein, in response to request to again obtain an exclusive copy of the memory block, the Home processor determines if the Owner processor is a sharer of the memory block, and if so, the Home processor sends an exclusive copy of the memory block to the Owner processor.

22. (Original) The method of claim 21, wherein the Home processor invalidates all other sharers when it sends an exclusive copy of the memory block to the Owner processor.

23. (Original) The method of claim 20, wherein, in response to request to again obtain an exclusive copy of the memory block, the Home processor directory determines if the Owner processor is a sharer of the memory block, and if not, the Home directory sends a Store Conditional Failure message to the Owner processor.

24. (Original) A distributed multiprocessing computer system, comprising:

a first processor that includes a memory block and a directory associated with said memory block that tracks the status of said memory block;

a second processor that includes a cache memory, and wherein said second processor is capable of requesting an exclusive copy of said memory block that is stored in said cache memory; and

wherein said second processor may displace the exclusive copy of said memory block prior to completing processing of said memory block, and said second processor transmits a signal to said first processor indicating that said second processor relinquishes exclusive control of said memory block but should remain a sharer of said memory block.

25. (Original) The distributed multiprocessing computer system of claim 24, wherein said second processor obtains an exclusive copy of said memory block by issuing a Load Lock instruction, and wherein the directory associated with the first processor indicates that said second processor has obtained exclusive control of said memory block.

26. (Original) The distributed multiprocessing computer system of claim 24, wherein said second processor is capable of executing multiple threads concurrently, and may displace data associated with a non-executing thread from its associated cache memory.

27. (Original) The distributed multiprocessing computer system of claim 24, wherein said second processor includes a register in which an address is stored representing the memory block exclusively obtained from said first processor, and wherein said second

processor compares the address of any displaced data with the address stored in said register.

28. (Original) The distributed multiprocessing computer system of claim 27, wherein the second processor asserts a Victim To Shared message if the address of any displaced data matches the address stored in said register.

29. (Original) The distributed multiprocessing computer system of claim 28, wherein the directory associated with the first processor indicates that said second processor has become a sharer of said memory block in response to said Victim To Shared message.

30. (Original) The distributed multiprocessing computer system of claim 24, wherein said second processor subsequently re-obtains an exclusive copy of said memory block from said first processor to complete processing of said memory block.

31. (Original) The distributed multiprocessing computer system of claim 30, wherein the second processor asserts a request to read, modify, and conditionally store said memory block to said first processor.

32. (Original) The distributed multiprocessing computer system of claim 31, wherein, in response to the request to read, modify, and conditionally store said memory block, the first processor determines if the second processor is a sharer of the memory block, and if so, the first processor sends an exclusive copy of the memory block to the second processor.

33. (Original) The distributed multiprocessing computer system of claim 32, wherein the first processor invalidates all other copies of said memory block when it sends an exclusive copy of the memory block to the second processor.

34. (Original) The distributed multiprocessing computer system of claim 31, wherein, in response to the request to read, modify, and conditionally store said memory block, the first processor determines if the second processor is a sharer of the memory block, and if not, the first processor sends a failure message to the second processor.

IX. EVIDENCE APPENDIX

None.

X. RELATED PROCEEDINGS APPENDIX

None.